

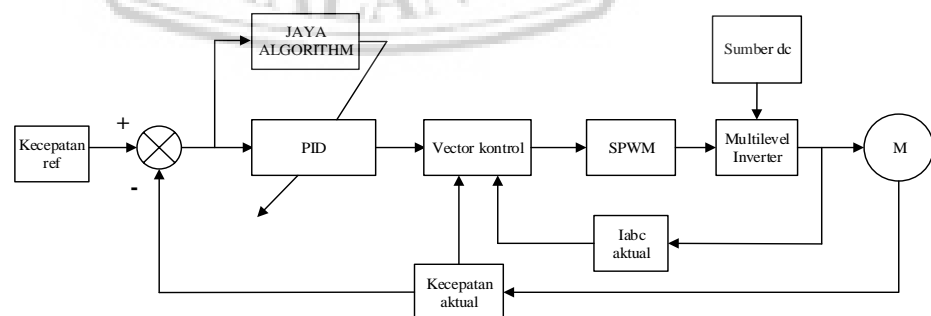
BAB III

METODELOGI PENELITIAN

Pada bab iii ini akan membahas mengenai perancangan pemodelan sistem dari pengendalian kecepatan motor induksi tiga fasa dengan *cascaded h-bridge multilevel inverter* menggunakan optimalisasi kontrol PID berbasis *Jaya Algorithm* dengan membuat simulasi rangkaian menggunakan program MATLAB Simulink R2017a. Perancangan simulasi rangkaian meliputi pemodelan motor induksi, pemodelan 3-level *cascaded h-bridge* (CHB) *multilevel inverter*, pemodelan *Phase Opposition disposition* (POD), pemodelan *vector control*, pemodelan *Kontroller proportional-integral-derivative* (PID), dan *Jaya Algorithm*.

3.1 Pemodelan Sistem

Pemodelan sistem simulasi rangkaian menggunakan motor 3 fasa dengan rotor sangkar, inverter 3 fasa, penyearah jembatan 3 fasa, dan kontrol kecepatan menggunakan PID yang di tuning menggunakan algoritma Jaya. Blok diagram kontroler kecepatan motor 3 fasa dengan menggunakan kontroler PID berbasis Jaya Algoritma dapat dilihat pada Gambar 3.1.



Gambar 3.1 Blok Diagram Sistem Pengendalian Kecepatan Motor Induksi

Tegangan sumber yang digunakan merupakan sumber tegangan DC , kemudian tegangan DC digunakan sebagai input sumber inverter 3 fasa yang akan

diubah kembali menjadi AC dengan amplitud dan frekuensi yang telah diatur sesuai dengan keinginan guna menjalankan motor induksi 3 fasa.

Actual speed motor induksi 3 fasa akan di umpan balikan ke kontroler PID dan lalu dibandingkan dengan masukan referensi kecepatan. Metode tuning parameter K_p , K_i , K_d dari kontroler PID ini akan dioptimalisasi menggunakan algoritma Jaya algoritma. Kecepatan dan arus aktual akan diumpan balikan juga ke *vector control*, kemudian didapat arus referensi. Arus referensi akan dimodulasikan bersama satu sinyal segitiga pada fasa R, S dan T yang memiliki sudut sama dengan frekuensi tinggi pada SPWM. Dari proses modulasi tersebut akan dihasilkan pulsa yang berfungsi sebagai *switching* pada *inverter* IGBT.

3.2 Pemodelan Motor Induksi Tiga Fasa Dalam Simulink

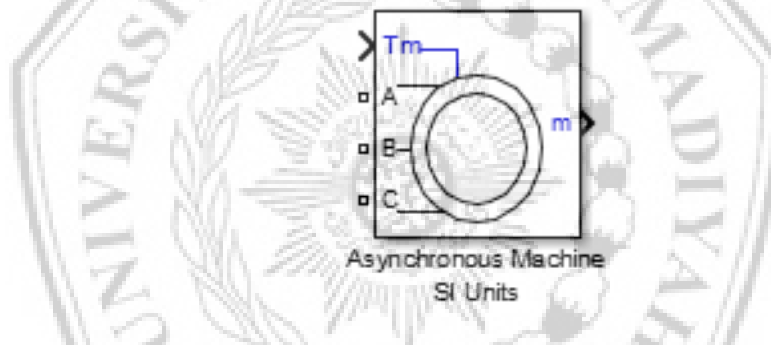
Model motor tersebut dirancang agar dapat disimulasikan dan mampu menampilkan beberapa indikator serta besaran yang merupakan titik ukur dari kinerja sebuah motor, sehingga dari hasil perancangan tersebut dapat mewakili kinerja motor induksi sebenarnya.

Cara lama dalam menganalisa suatu motor induksi tiga fasa digunakan model trafo dengan mengasumsikan dimana tegangan dari sumber yaitu sinusoidal dan *steady state*. Dalam praktiknya, terdapat permasalahan tegangan sumber yang tidak sinusoidal dan terjadi perubahan beban. Sebab itu digunakan model berbeda supaya lebih fleksibel dalam analisa motor induksi menggunakan analisa dalam koordinat d-q-n. Model d-q bisa menganalisa motor induksi dalam kondisi : *transient*, *steady state*, dan perubahan beban. Persamaan 2.1-2.14 dalam motor dengan koordinat *dq* diatas, pemodelan motor induksi yang akan digunakan adalah tipe motor induksi rotor sangkar tupai (*squirrel cage*) yang telah disediakan oleh *MATLAB Simulink* yang ditunjukkan pada Gambar 3.2. Motor induksi yang dipakai dalam merujuk spesifikasi yang sudah ada dan dapat dilihat pada tabel 3.1.

Tabel 3.1 Data Parameter Motor Induksi tiga Fasa *MATLAB Simulink*

Rated Power / Daya	20HP/14920W
Rated Voltage	460 V
Rated Frequency	60 Hz
Rated Speed	1760 RPM
Pole	2

Pemodela dari motor induksi tiga fasa (motor asinkron) yang digunakan di *SimPowerSystemsTM library*, ditunjukkan pada Gambar 3.2

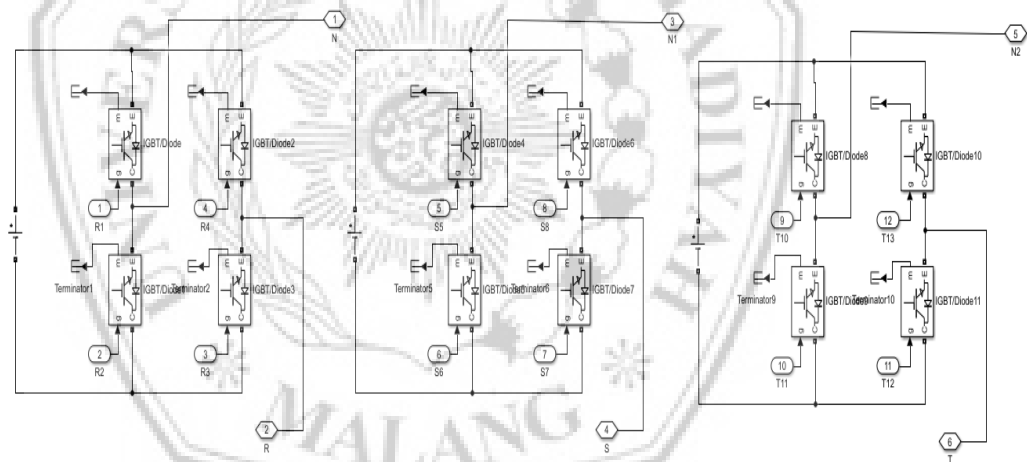


Gambar 3.2 Pemodelan Motor Induksi pada *MATLAB Simulink*

Parameter yang dipakai dalam model motor induksi antara lain, nominal power (P_n) = 1.492e+04 VA, line to line voltage (V_n) = 460 Vrms, frekuensi (f_n) = 60 Hz, stator resistance (R_s) = 0,2761 Ohm, stator inductance (L_{is}) = 0.002191 Henry, rotor resistance (R_r') = 0.1645 Ohm, rotor inductance (L_{ir}') = 0.002191 Henry, mutual inductance (L_m) atau magnetizing inductance = 0.7614 Henry, kombinasi koefisien momen inerti mesin (J) = 0.1 kg.m², friction factor (F) = 0,01771 n.m.s, dan jumlah kutubnya atau pole (p) = 2 pasang.

3.3 Pemodelan *cascade H-Bridge Multilevel Inverter*

Multilevel Inverter adalah rangkaian inverter yang terdiri dari beberapa *inverter* jembatan penuh pada masing masing fase (*cascade*). *Multilevel inverter* disusun secara seri dengan sumber tegangan DC. Setiap unit *H-Bridge* memiliki sumber dc sendiri dan untuk motor induksi adalah unit baterai. Keluaran AC dari tiap *full bridge inverter* yang berbeda dihubungkan secara seri, sehingga menghasilkan sinyal tegangan yang dihasilkan dari penjumlahan dari setiap keluaran *inverter*. Pada *3-bridge multilevel inverter* terdapat 3 rangkaian *full bridge inverter* dengan 1 sumber DC pada setiap fasanya. Pemodelan *inverter* 3 tingkat mirip dengan 5 tingkat *inverter*. Perbedaannya adalah pada banyaknya sinyal pembawa. Pada 3 tingkat *inverter* ada 2 sinyal pembawa. Pemodelan pada *3-bridge multilevel inverter*. Multilevel nverter ditunjukkan pada Gambar 3.3.



Gambar 3.3 Pemodelan *3-bridge multilevel nverter* Tiga Fasa pada *MATLAB Simulink*

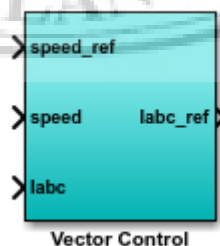
Parameter yang digunakan yaitu nilai *snubber resistance* (R_s) = 1000 ohm, *snubber capacitance* (C_s) = inf mendapatkan snubber yang bersifat resistif, untuk *internal resistance devais* yang dipilih (R_{on}) = 1e-3 ohm, dan *forward voltage* 0.8 volt.

3.4 Pemodelan *Vector Control*

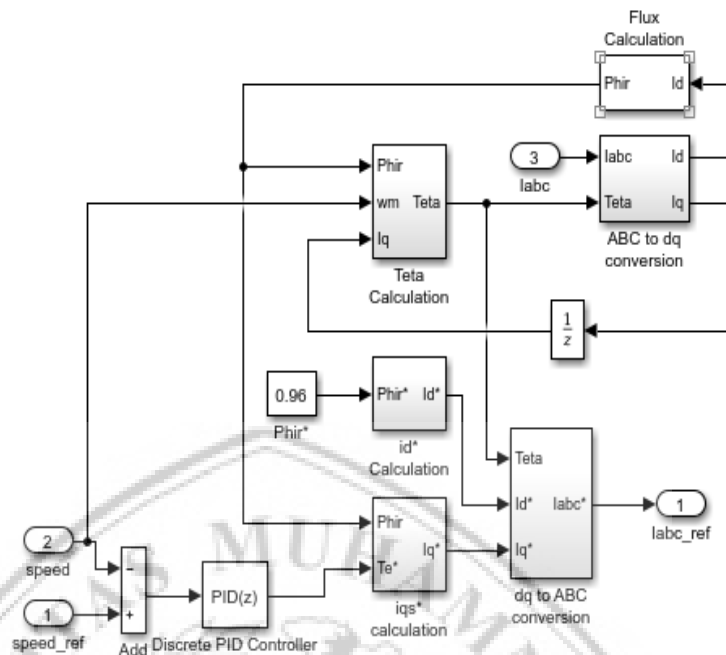
Vector control menghubungkan motor induksi dengan IGBT *inverter*. *Vector control* mengeluarkan sinyal PWM yang dipakai sebagai pengatur frekuensi dan amplitude *inverter*. Frekuensi yang telah diatur sesuai kebutuhan akan digunakan untuk menggerakkan atau memutar motor induksi tiga fasa sesuai kecepatan referensi.

Vector control merupakan sebuah cara pengontrolan *coil field* dalam motor AC, dari sistem *couple* akan diubah ke *decouple*. Dengan sistem ini, arus penguatan dan arus beban motor akan diatur secara terpisah seperti halnya pada motor DC. Penerapan *flux vector* membutuhkan simulasi stator, rotor dan torsi. Untuk dapat mempermudah simulasi pada motor induksi dari kondisi transien menuju pada kestabilan, maka koordinat abc motor harus diganti kedalam bentuk model bayangan/besaran *vector* kebentuk *direct quadrature* (dq). Proses berubahnya koordinat abc menjadi koordinat *direct quadrature* (dq) dengan penggabungan persamaan transformasi Clarke dan Park yang ditampilkan pada persamaan 2.24-2.33 untuk mempermudah dalam pemodelan serta simulasi yang akan dijalankan.

Rangkaian dan model *vector control* yang digunakan di pada *SimPowerSystems™ library*, ditunjukkan pada Gambar 3.4 dan Gambar 3.5.



Gambar 3.4 Pemodelan *Vector Control* pada *MATLAB Simulink*

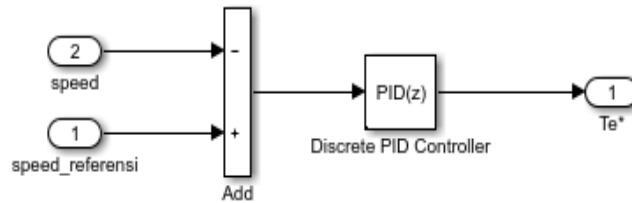


Gambar 3.5 Blok Rangkaian Transformasi *Vector Control* pada *MATLAB Simulink*

Pada rangkaian *vector control* terdiri dari beberapa bagian seperti pada Gambar 3.5 di atas, antara lain :

3.4.1 Kontroler PID (*Proportional Intergral Derivative*)

Dalam rangkaian *vector control* menggunakan dua buah acuan yaitu : kecepatan referensi dan aktual, dengan hasil selisih dari kecepatan referensi motor dan kecepatan aktual motor yang akan menjadi *input* dari *PID speed control*. *PID* sendiri ditunjukkan pada persamaan 2.44, 2.45, dan 2.46. Dalam *PID speed control* terdapat rangkaian *Proportional Intergral Derivative* (*PID*) indikator *gain* (penguat) K_p , K_i , dan K_d yang akan dioptimasi menggunakan algoritma Jaya. Blok *PID* pada matlab ditunjukkan pada Gambar 3.6.

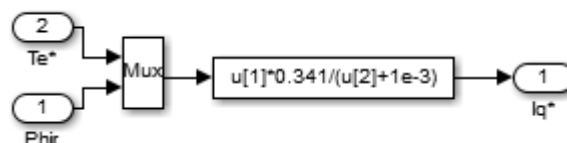


Gambar 3.6 Rangkaian Kontroler *Proportional Integral Derivative* (PID) pada *MATLAB Simulink*

Gain (penguat) K_p digunakan untuk respon *rise time* dan *setting time* pada saat motor *start*. *Gain* K_i disambungkan ke *discrete time integrator* yang berperan untuk mengurangi nilai *overshoot* pada sinyal *output* motor. *Gain* K_d juga dihubungkan ke *discrete derivative* berguna dalam mempertahankan kecepatan motor pada kecepatan referensi. Perhitungan nilai dari parameter K_p , K_i , dan K_d disaturasikan ke *torque electromagnetic* (T_e^*).

3.4.2 I_{qs}^* Calculation

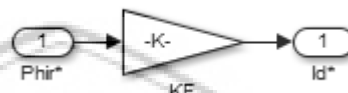
T_e^* keluaran dari PID akan menjadi salah satu *input vector control* yang dihubungkan pada *iqs* calculation*, adapun persamaan *iqs* calculation* seperti ditunjukkan pada persamaan 2.40. *iqs* calculation* disini berguna menghasilkan arus i_q^* , yaitu komponen arus *q-axis* referensi pada stator dengan *input* torsi referensi menghasilkan *output* dari PID kontroler dan fluks rotor (ϕ_{hir}). Kemudian hasil dari *iqs* calculation* digunakan sebagai masukan untuk blok *dq to ABC conversion*. *Iqs calculation* ditunjukkan pada Gambar 3.7.



Gambar 3.7 Rangkaian *iqs* calculation*

3.4.3 Ids* Calculation

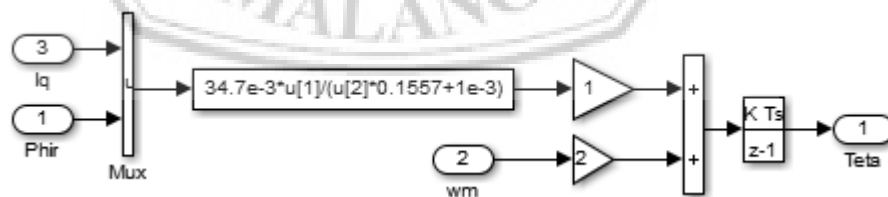
Pada persamaan 2.41 merupakan persamaan dari i_{ds}^* calculation. i_{ds}^* calculation berguna menghasilkan arus i_{ds}^* yaitu komponen arus d-axis referensi pada stator, dengan *input* fluks rotor referensi (ϕ_{ir}^*). Hasil dari i_{ds}^* calculation digunakan sebagai input pada blok dq to ABC conversion. Blok Id calculation ditunjukkan pada Gambar 3.8.



Gambar 3.8 Rangkaian i_{ds}^* calculation

3.4.4 Teta Calculation

Teta calculation berfungsi menghitung nilai sudut antara rotor dan stator dalam fungsi sudut radian. Dalam mencari nilai dari *teta calculation* tersebut ditunjukkan pada persamaan 2.42. Perlu adanya parameter i_{qs} dan ϕ_{ir} serta rotor mechanical speed (ω_m) atau kecepatan *input*. Selisih antara i_{qs} dan ϕ_{ir} akan disambungkan ke *function block parameters* sebagai perhitungan hasil dari keluaran *teta calculation*, lalu menuju ABC to direct-quadratic (dq) conversion. Blok *Teta Calculation* ditunjukkan pada Gambar 3.9.

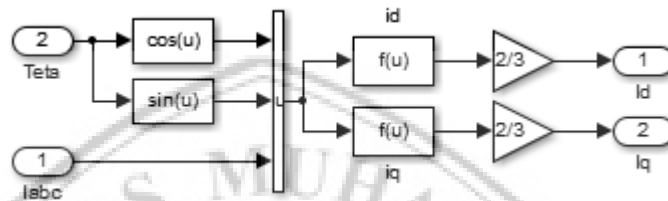


Gambar 3.9 Rangkaian *teta calculation*

3.4.5 ABC to dq conversion

ABC to dq conversion berfungsi merubah arus I_a , I_b , dan I_c (aktual) yang terukur di stator menjadi arus dq pada bidang ortogonal. Dalam bidang ini dapat didapat nilai fluks rotor, torsi motor, dan *theta* yaitu menemukan *phase angle* dari

rotor fluks. *Input* dari ABC to dq conversion adalah sudut *theta* dan arus ABC aktual. Blok transformasi ABC to dq ini membutuhkan pergeseran sudut antara *direct* dan *quadratic teta calculation* dalam fungsi *sinus* atau *consinus* yang mendapatkan keluaran *iq* dan *id* seperti pada persamaan 2.34 dan 2.35. Dimana arus *id* akan masuk ke *flux calculation*. Blok ABC to dq conversion ditunjukkan pada Gambar 3.10.



Gambar 3.10 Rangkaian ABC to dq conversion

3.4.6 Flux Calculation

flux calculation berfungsi menghitung fluks rotor, hasilnya adalah nilai *phir* yang diukur dan digunakan menghitung i_{qs} setiap saat. Pada blok ini, yang terpenting yaitu *discrete transfer fuction*. *Transfer function* yang muncul karena perubahan yang terjadi tiap saat merupakan orde satu dengan periode $T = 0.1557s$ yang dipakai untuk mengintegrasikan perkalian *id* dan L_m menuju ke ABC to *direct-quadratic* (dq) conversion. Blok *flux calculation* dapat dilihat pada Gambar 3.11.

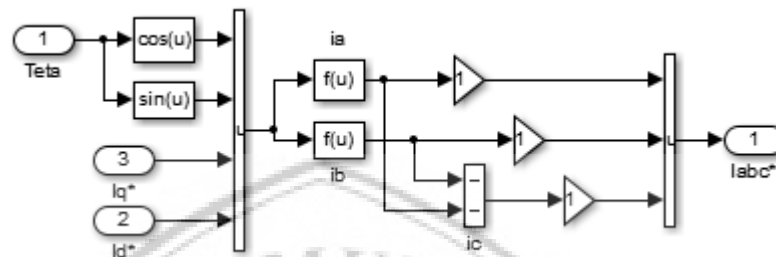


Gambar 3.11 Rangkaian *flux calculation*

3.4.7 Dq to ABC conversion

dq to ABC conversion merubah arus dq menjadi arus abc referensi (I_{abc}^*). Arus abc (I_{abc}^*) dan arus abc aktual (I_{abc}) akan menjadi *input* pada *current regulator* untuk menghasilkan pulsa. Adapun masukan *dq to ABC conversion* yaitu

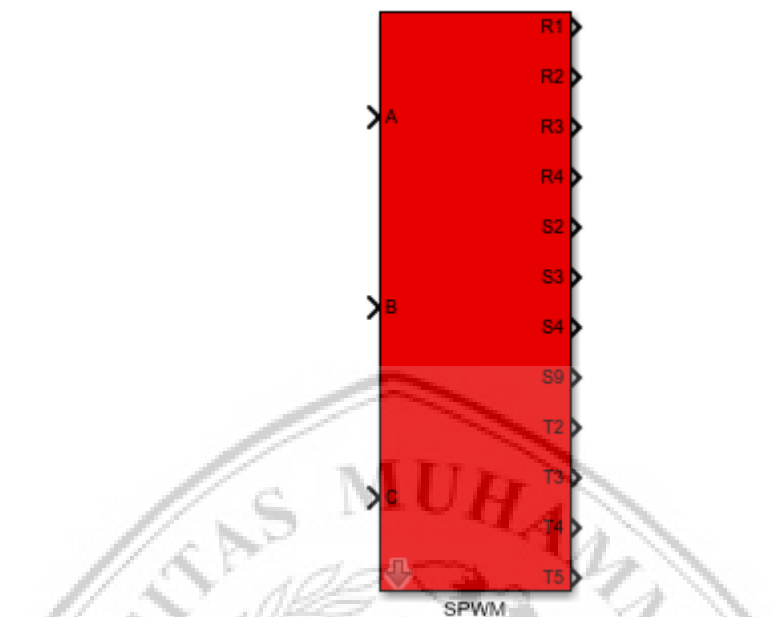
sudut θ , arus i_q^* , dan arus i_d^* . Blok *dq to ABC conversion* membutuhkan pergeseran sudut antara *direct* dan *quadratic teta calculation* dalam fungsi *sinus* atau *consinus* yang menghasilkan arus I abc seperti ditunjukkan pada persamaan 2.37 sampai 2.39. Blok *dq to ABC conversion* ditunjukkan pada Gambar 3.12.



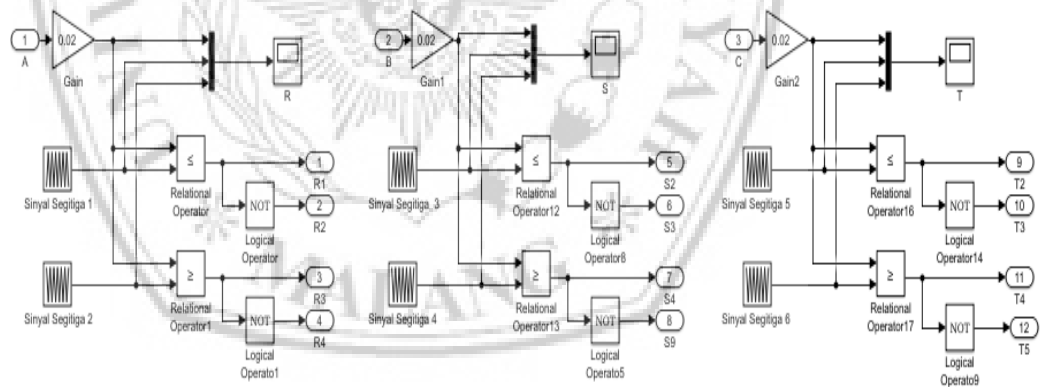
Gambar 3.12 Rangkaian *dq to ABC conversion*

3.5 Pemodelan *Sinusidal Pulse Width Modulation* (SPWM)

THD (*total harmonic distortion*), merupakan suatu masalah yang timbul dalam proses konversi *dc to ac* pada *inverter*. Dimana THD atau harmonisa menggambarkan pembentukan sinyal sinusidal dan nilainya adalah hasil nilai perkalian dari frekuensi dasarnya. Harmonisa membuat gangguan atas sistem hingga kerusakan pada piranti listrik. Dari permasalahan tersebut, maka di perlukan suatu metode untuk mengurangi THD, salah satu metode tersebut adalah dengan pengaturan lebar pulsa menggunakan SPWM. *Sinusidal pulse width modulation* (SPWM) merupakan salah satu metode yang digunakan pada operasi switching inverter. Pada metode sinus segitiga ini, sinyal gating dihasilkan dengan membandingkan gelombang sinusida referensi dengan gelombang segitiga pembawa frekuensi F_r dan F_c . Pemodelan rangkaian dari SPWM dapat ditunjukkan pada Gambar 3.13 dan Gambar 3.14.



Gambar 3.13 Pemodelan *SPWM* pada *MATLAB Simulink*



Gambar 3.14 Rangkaian *SPWM* (*Sinusoidal pulse widtht modulation*) pada *MATLAB Simulink*

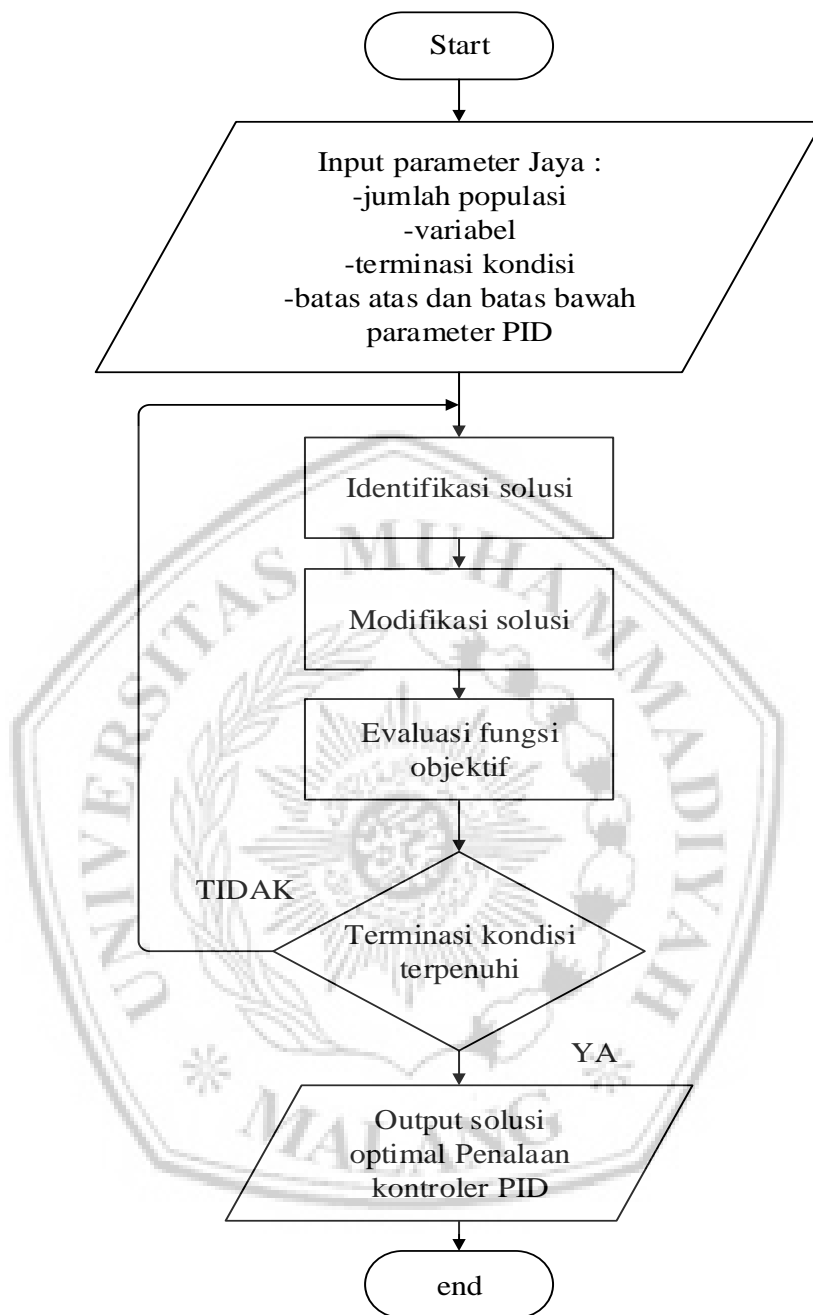
Rangkaian pada Gambar 3.14 akan menghasilkan *pulse* sebagai masukan gate transistor IGBT pada *cascade 3-bridge multilevel inverter* yang dihubungkan dengan motor induksi 3 fasa.

3.6 Penalaan Parameter PID Menggunakan *Jaya algorithm*

Jaya adalah salah satu algoritma optimasi baru yang kuat namun sederhana. Jaya digolongkan sebagai algoritma dengan parameter kurang atau tidak memerlukan parameter yang khusus, seperti pada algoritma TLBO (*Teaching learning based optimization*) [12]. Jaya algoritma hanya memerlukan 3 inisialisasi yaitu jumlah populasi, jumlah variabel, dan kondisi terminasi. Pada Jaya algoritma hanya bergantung pada 1 buah rumus pada persamaan 2.47 dalam pencarian solusi nilai terbaiknya.

Dalam penalaan parameter PID memerlukan fungsi objektif yang berperan penting sebagai indeks performa yang dapat menyusutkan nilai-nilai maksimum overshoot dari respon step sistem. Sebuah sistem kontrol yang optimal, akan dapat meminimalkan indeks performansi. Indeks performansi merupakan suatu fungsi yang nilainya menunjukkan seberapa bagus kerja sistem. Ada beberapa fungsi objektif yang digunakan dalam pengujian sistem untuk melihat mana yang terbaik. Fungsi tujuan untuk pengujian sistem antara lain menggunakan SSE (sum of square error), ISE (*integral of the square of the error*), IAE (*integral of absolute magnitude of error*) dengan menggunakan persamaan pada 2.48, 2.49, dan 2.50. Indeks performansi diatas digunakan untuk meminimalkan *overshot*, *settling time*, *steady state error*, dan referensi *tracking error* [13]. Fungsi tersebut akan digunakan untuk indeks performansi dari kontroler PID menggunakan algoritma optimasi.

Algoritma Jaya implementasinya dalam penalaan parameter PID bertujuan untuk mendapatkan nilai dari parameter PID yaitu k_p , k_i , dan k_d . Pada Penalaan parameter kontroler PID menggunakan *Jaya Algorithm* dapat dilihat pada Gambar 3.15.



Gambar 3.15 Flowchart penalaan parameter PID dengan Jaya Algoritim

3.7 Pengujian Sistem

Pengujian yang dilakukan yaitu untuk mengetahui kinerja dari sistem kendali motor induksi dengan optimasi PID berbasis Jaya algorithm dan memastikan hasil perancangan sesuai dengan yang diinginkan. Adapun beberapa pengujian yang dilakukan sebagai berikut :

1. Pengujian tiga fungsi objektif SSTE, ISE, dan IAE dengan algoritma jaya dan FPA. Beban yang digunakan bernilai 0 Nm. Pengujian ini dilakukan dengan tiga fungsi indeks diatas secara bergantian pada jaya algoritma dan FPA.
2. Pengujian multilevel inverter, pengujian ini dilakukan dengan menguji THD, sinyal keluaran tegangan dan arus pada inverter konvensional dan multilevel inverter. Dengan beban bervariasi 0 Nm, 20 Nm, dan 50 Nm
3. Pengujian Kecepatan menggunakan PID konvensional dan PID Jaya algoritma dengan menggunakan *multilevel inverter*. Beban yang digunakan adalah beban yang bervariasi yaitu 0 Nm, 20 Nm dan 50 Nm.
4. Pengujian Kecepatan menggunakan PID konvensional dan PID Jaya algoritma dengan menggunakan *multilevel inverter* dengan beban berubah pada $t=1$ second. Besar beban saat berubah yang digunakan bernilai 50 Nm.